7/25/22
Group 55
Daniel Joseph
Ethan Riemer
Project Step 3 Final

https://cs340group55.herokuapp.com/

Peer Feedback (Step 1)

Musa Harb:

·**Does the overview describe what problem is to be solved by a website with DB back end?**

Yes. The draft describes the rapid growth of the film industry and increased competition as more studios enter the industry. The increased demand by the major entertainment companies and reduced supply in recent years has created a need to create an efficient tracking system of the production process. The draft proposes a database that is available to all film producers that can be used by all film and television shows creators to track production needs including studio orders, products, vendors, purchases, and sales reps.

·**Does the overview list specific facts?**

Yes, the overview provides facts about the growth of the film industry and projected future growth. Currently the film industry generates 95.45 billion in revenue annually and it is expected to grow by 7.2% annually over the next few years to 169.68 billion.

·**Are at least four entities described and does each one represent a single idea to be stored as a list?**

There are 6 entities that each represent a single idea and are stored as a list. The entities include:

§ Sales Reps

§ Orders

§ Products

§ Vendors

§ Productions

§ Studios

·**Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?**

Yes, each entity's purpose is described.

All entity's attributes are listed including type and constraints. I suggest adding the size of the VARCHAR datatypes as well to each entity's description.

All relationship types are described in each entity's description.

·**Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?**

Yes, All 1:M relationships are correctly formulated in the ERD and described thoroughly in each entity's description. There are six 1:M relationships.

There is one M:M relationship between the Orders entity and the Products entity. The M:M relationship is correctly formulated by an intersectional table in the ERD, Orders_has_Products. The intersectional table has a 1:M relation with Orders and 1:M relationship with Products

·**Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**

Yes, there is consistency in naming between overview and entity and attributes. The entities names are all plural and the attributes are singular. The entities names are capitalized. All attributes are listed using camel case naming convention.

I suggest not using space in entities' names and changing "Sales Reps" to salesRep. Also, the intersectional table "Orders_has_Product" is in snake case naming convention. I suggest changing it to "ordersHasProducts".

Alex Fren:

Does the overview describe what problem is to be solved by a website with DB back end?

Yes - it outlines the growth in the broader industry that this business fits into, and how that has impacted this companies' growth. The list multiple factors that have played into this, and what the expected impact will continue to be projected into the future.

Does the overview list specific facts?

The overview lists many specific facts in illustrating the problem that the website addresses in the industry at large, but not as many for this specific company and its database. They do mention they are tracking hundreds of orders, products, vendors, etc, but these numbers could be more specific for each entity to really highlight the issue.

Are at least four entities described and does each one represent a single idea to be stored as a list?

Yes, they are Sales Reps, Products, Vendors, and Studios, in addition to transactional and linking tables.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?

Yes - purpose is clearly outlined for each table, and relationships between each table are described. Constraints like FK, NOT NULL, etc are correctly used.

In the linking table, Orders_has_products, there are two keys denoted as a PK, which I don't believe is possible. I think these are meant to be FK, with a third attribute serving as a PK.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?

I believe all 1:M are correctly formulated, with the exception of productions and orders, which describes the relationship as 1:M in both directions. There is a M:M relationship, between Orders and Products, with a linking table between them. ERD matches relationships described in the overview.

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

Mostly consistent, though the linking tables (Orders_has_Products) use snake case for the table name and attributes where all else use Camel Case. All tables are plural and attributes singular. Names are consistent from overview to ERD.

Zachary Baker:

Does the overview describe what problem is to be solved by a website with DB back end?

Yes. This seems plausible to be a website that tracks orders of production equipment on behalf of production companies, allows both the FIFTYFIVE company to know what orders have been placed for which items from which studio.

Does the overview list specific facts?

Yes. The facts are mostly about the industry more than about the company's problem to be solved but it is stated that this website will address the needs of a company with $55m in annual revenue and that it will service production sizes as small as student films up to major feature films.

Are at least four entities described and does each one represent a single idea to be stored as a list?

Yes. Sales Reps, Products, Vendors, and Productions are all entities and they all appear to represent single ideas to be stored in the DB.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?

Yes. The descriptions of each entity are clear, relationships are well defined (including FKs) and data types are used on all attributes. Some things that can be improved, I think, would be adding more specific constraints to data types like VARCHAR (how many characters?), updating price/cost from INT to DECIMAL (or a FLOAT, just something that can track financial amounts that are not even dollars), and consider adding more granular information to address as it may be difficult to store addresses in a searchable way having it all in a single attribute, and this could also lead to data duplication between the different entities that both use addresses (such as Productions and Studios).

Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?

Yes 1:M relationships seem clearly stated and correctly formed with PK/FK relationships. I believe there is one M:M relationship between Orders:Products and that a Order_has_Products table has been inserted as the intersection table. However, I think the Orders_has_Products may need different attributes. I think the two FK attributes are orderID and productID and that Orders_has_Products is missing its own PK for intersection ID. Currently, it looks like two product IDs and I don't think that is right. Not sure if an intersection ID needs its own unique PK but I don't see any reason why you couldn't have each intersection that is mapped have it's own ID for data management purposes

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

Yes. Great consistency between capitalization/naming between entities and attributes and pluralization. An improvement could be to use spaces vs. underscore consistently, for example, "Sales Reps" might be better as either "SalesReps" (to match the convention used in their attributes that would otherwise have spaces like "termsCode", "contactEmail", and "retailPrice", for example) or either "Sales_Reps" (to match the convention from the "Orders_has_Products" table).

Jonah Tang:

Does the overview describe what problem is to be solved by a website with DB back end?

Yes. The outline details the surge in growth of the film industry, which necessitates the need for a service to be used by production companies in order to streamline their developments. The service tracks and records production items purchased by the production company, such that the company and studios may focus their efforts elsewhere.

Does the overview list specific facts?

Yes. The overview includes facts about the growth of the film industry, including figures such as an expected revenue generation of $95.45 billion this year and a 7.2% annual growth over an 8 year period.

Are at least four entities described and does each one represent a single idea to be stored as a list?

Yes. The entities are Sales Reps, Orders, Products, Vendors, Productions, and Studios.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?

Yes. Each entity's purpose is described, and their attributes properly detail the datatypes and constraints. The relationships and implementations are also described for each entity.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?

Yes. The 1:M relationships are correctly formulated. There is one M:M relationship, between the Products and Orders entities. This relationship is implemented through the intersection table Orders_has_Products. The ERD presents a logical view of the database and well-illustrates the relationships between entities, including the intersection table.

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

Naming is largely consistent with regard to plurality and capitalization. However, the "Sales Reps" entity's name includes a space, whereas "Orders_has_Products" utilizes underscores; one of the names could be changed to maintain consistency.

Actionable Improvement Steps (Step 1)

- Added VARCHAR length for accurate data type storage.
- Fixed entity table naming convention to camelCase and standardized attribute naming conventions across all entities.
- Added specific numerical data points to database description to address scalability concerns.
- Fixed intersection table to better align with SQL best practices, this includes a proper unique Primary Key and proper use of Foreign Keys.
- Updated Product cost, retail pricing, and total order amount data type to DECIMAL(19,2) to accurately reflect currency.

- Added complete postal address details to Studios, Productions, and Vendor entities to prevent instances of duplicate data.
- Added TermsCode entity table to optimize database efficiency and maintain consistency.

Peer Feedback (Step 2)

Carl Towner:

•Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

Yes.

•Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

There are some minor naming inconsistencies regarding zip vs zipCode and a length of one VARCHAR attribute (see below for screenshot)

•Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

Yes the schema is easy to read and there are no lines crossed.

•Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

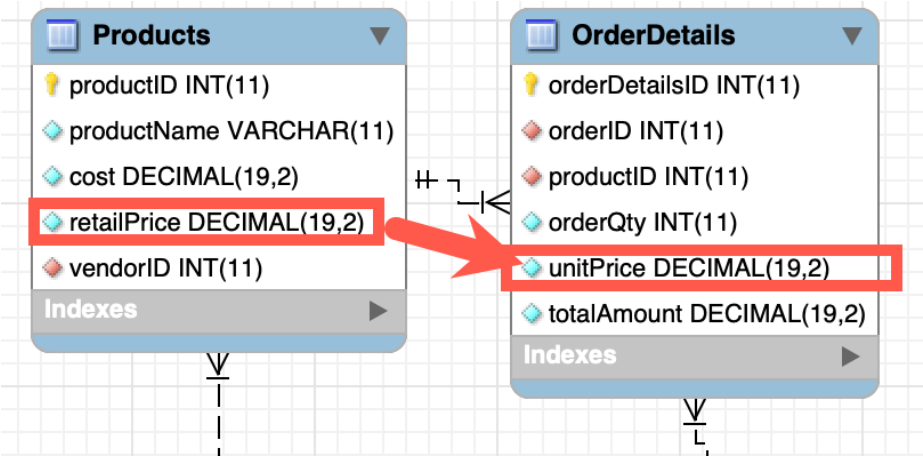
Yes.

- The OrderDetails table has orderID and productID as FKs.
- The Orders table has productionID, termsCodeID, and salesRepID as FKs.

•Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?

I could only find one potential issue... Products.retailPrice = OrderDetails.unitPrice. It's a good idea to record the unit price within the OrderDetails table...especially if the price of those screws changes so that if someone wants to return a few for a refund that you don't end up paying them more than they paid for them. However, I don't see anything enforcing

this dependency within the database itself. I'm sure you'll take care to appropriately manage this in the application. Likewise I don't see any code within the amountTotal attribute to compute the orderQty x unitPrice.



•Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)

SQL loaded without errors.

The drop table statements at the beginning of the .sql file are not needed since CREATE OR REPLACE is used for all tables.

•In the SQL, are the data types appropriate considering the description of the attribute in the database outline?

Yes.

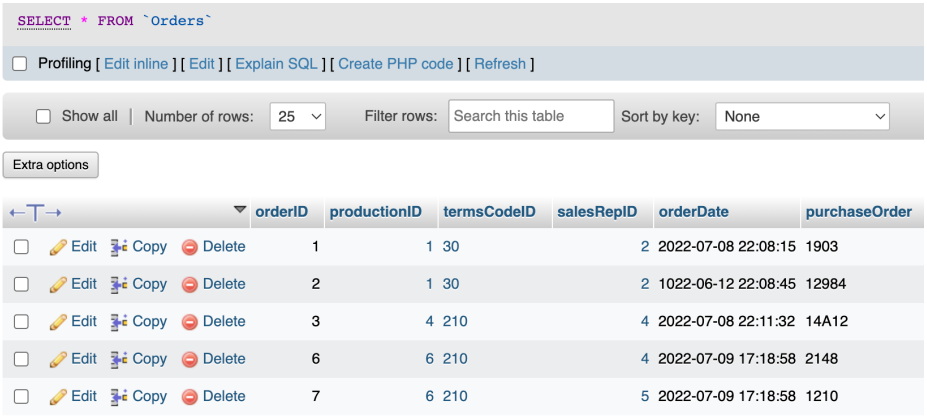
•In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?

ON DELETE CASCADE is used in all instances.

I think this may lead to some unintentional data loss. For example, let's start with seeing all the orders our sales reps have been involved with...

name	orderID	orderDate	SUM(`OrderDetails`.`totalAmount`)
Mike Row	1	2022-07-08 22:08:15	99.00
Mike Row	2	1022-06-12 22:08:45	100.00
Amanda Stone	3	2022-07-08 22:11:32	80.00
Dan Joseph	4	2022-07-09 17:17:14	749.00
Dan Joseph	5	2022-07-09 17:17:14	124.00
Amanda Stone	6	2022-07-09 17:18:58	400.00
John Francis	7	2022-07-09 17:18:58	422.00

Now Dan Joseph has updated his LinkedIn and is advertising that he's the top seller...subsequently he gets a better offer and moves on. What happens to the Orders 4 and 5 when we delete Dan Joseph?

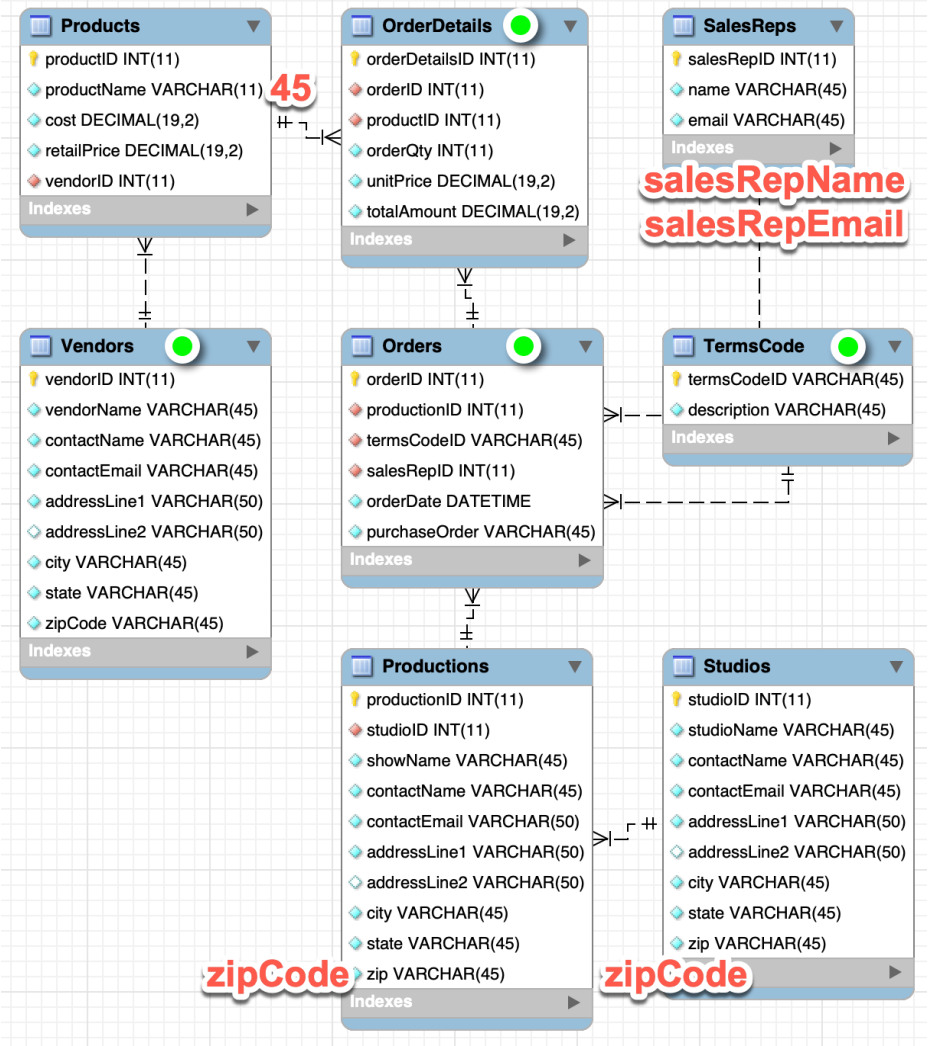


Orders 4 and 5 are gone because we deleted Dan Joseph as a salesRep. I don't know what the right answer is here, but this probably isn't what we want. Perhaps we can add an attribute to indicate whether a salesRep is currently an employee? Change the CASCADE option? Not sure.

A similar risk is present when deleting a vendor.

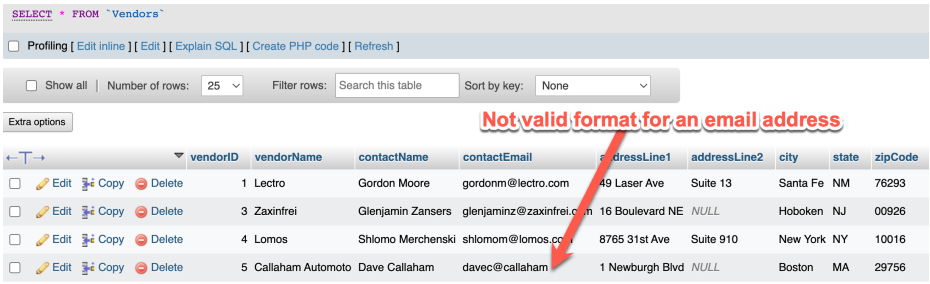
•In the SQL, are relationship tables present when compared to the ERD/Schema?

I ran a reverse engineer operation on the .sql file using MySQLWorkbench and got the following ERD. Inconsistencies are minor and noted in red.

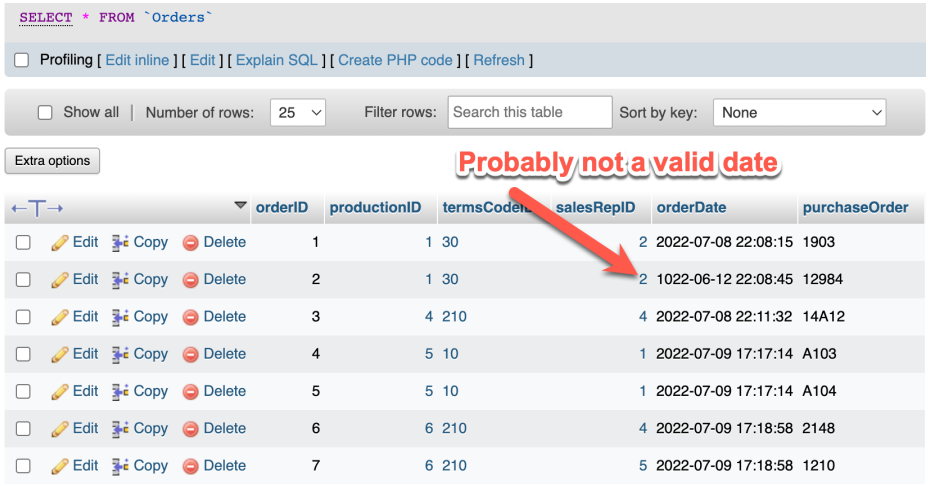


•In the SQL, is all example data shown in the PDF INSERTED?

Yes the data is inserted. Regarding emails, it might be better to increase the size, for example VARCHAR(319), and include a validity check so that improperly formatted emails do not get stored. Emails should also have a unique constraint on them.



May want to add some constraints on the date as well.



Some fields are too small.

```
-- Products Table
--
CREATE OR REPLACE TABLE `Products` (
        `productID` int(11) NOT NULL AUTO_INCREMENT UNIQUE,
        `productName` varchar(11) NOT NULL,
        `cost` decimal(19, 2) NOT NULL,
        `retailPrice` decimal(19, 2) NOT NULL,
        `vendorID` int(11) NOT NULL,
        PRIMARY KEY (`productID`),
        FOREIGN KEY (`vendorID`) REFERENCES Vendors(`vendorID`) ON DELETE CASCADE
    );
--
-- Products Data
--
INSERT INTO `Products` (
        `productID`,
        `productName`,
        `cost`,
        `retailPrice`,
        `vendorID`
    )
VALUES (1, 'Wireless Sticks', '139.09', '499.99', 1),
    (2, 'Blomight', '1298.12', '4999.99', 4),
    (3, 'Machanio', '45.12', '103.45', 5),
    (4, 'Screws', '0.01', '1.00', 3),
    (5, 'Mustache Feet', '421.99', '505.10', 1),
    (
        6,
        'Grip to Ground Adapter',
        '249.99',
        '349.99',
        5
    );
```

Too small

Wireless St

Some tables have a single "name" field and I'd recommend splitting that up into something like contactFirstName, contactLastName because otherwise you could have a bigger challenge when searching for people within those tables.

Musa Harb:

Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

There is no schema, but there is an ERD model that contains all details that would be included in the schema. The detailed ERD present a physical model that follows the database outline.

Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

There are consistency in naming between the overview and the ERD, but there are few mismatches:

·In the database outline salesRepName and salesRepEmail attributes in the SalesReps entity are listed as name and email. I suggest changing attributes names in the outline to match the ERD

·productName datatype is listed as VARCHAR(11) in the outline while it is listed as VARCHAR(45) in the ERD. I suggest changing the data type size in the outline to match the ERD

·The zipCode attribute is listed as zip in the Studios entity in the outline. I suggest changing the attribute name in the overview to match the ERD

Entities are all plural and attributes are singular. All entities names are capitalized.

Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

The detailed ERD (although there is no diagram labeled as schema the ERD has schema level details) is easy to read, clear, and all relationship lines not crossed.

Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

Yes, there is a M:N relationship between the Orders and Products entities which is formed correctly with OrderDetails intersection table that has a M:1 with Orders and Products. The relationships are facilatied with 2 FKs, OrderID and ProductID which are included in the OrderDetails table.

Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?

All sample data indicate normalized data base design except for potential update anomaly in the Productions table.

I suggest adding another table for productions staff to avoid update anomaly. For example, if you had one contact name responsible for multiple shows and you need to update the contact information for that person you will have to update each row that includes that person in the Productions table.

Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)

Yes, the file is syntactically correct.

In the SQL, are the data types appropriate considering the description of the attribute in the database outline?

Yes all data types included in the SQL file are appropriate and match the description of the attribute in the database outline.

In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?

As mentioned above, there was no schema in the draft; however the primary and foreign keys are correctly defined compared to the detailed ERD and the database outline.

CASCADE ON DELETE is included the Productions table for the studioID FK, in the Products table for the vendorID FK, on the Orders tables on the productionID, salesRepID, and termsCodeID FKs, and the OrderDetails table on the ordered and productID FKs.

I suggest changing the CASCADE operation on the Orders table to ON DELETE RESTRICT, since the Orders table is recording transactions, I think it will be better if all delete and update operations are done directly on the Orders table. Or change it to ON

DELETE SET NULL that way you set the deleted FK value to NULL without deleting the entire order (row) in the Orders tables.

In the SQL, are relationship tables present when compared to the ERD/Schema?

Yes, all relationship tables are present in the SQL as described and outlined in the ERD.

In the SQL, is all example data shown in the PDF INSERTED?

Yes, all example data listed in the PDF are included as INSERT queries in the SQL file.

Erik Hoffman:

- **Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?**

The schema is labeled as the ERD but their is no ERD present but based on the description the schema looks good.

- **Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**

naming between outline and schema look good but again there is no ER diagram.

- **Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?**

Yes.

- **Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?**

Yes intersection tables are formed properly.

- **Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?**

Saw no non-normalized issues.

- **Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)**

Yes when run on phpMyAdmin all queries were ok.

- **In the SQL, are the data types appropriate considering the description of the attribute in the database outline?**

data types were appropriate based on the descriptions of the attribute.

- **In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?**

Yes.

- **In the SQL, are relationship tables present when compared to the ERD/Schema?**

Yes.

- **In the SQL, is all example data shown in the PDF INSERTED?**

Yes.

Actionable Improvement Steps (Step 2)

- Added separate ERD and Schema to report for clarity.
- Fixed naming, data types, and data type size inconsistencies between database outline and schema.
- Updated Email data types to unique attributes.
- Changed ON DELETE operations to NULL to prevent data loss.
- Fixing CASCADES fixed deletion anomalies.
- Added additional attributes to TermsCodes entity for end user readability.
- Updated outline to match updated Schema

Peer Feedback (Step 3)

Musa Harb:

Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

Yes, the UI utilizes SELECT for all tables in the schema. The OrderDetails table is joined with the Order table in the same SELECT.

·**Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?**

Yes, there are three search/filter with a dynamically populated list of properties, “Show All Productions By Studio”, “Show All Orders by Production”, and “Show all Products by Vendor”.

·**Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.**

Yes, the UI utilizes an INSERT for every table in schema.

·**Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).**

Yes adding an order, adds a row to the OrderDetails intersection table that facilitates the M:M relationship between Orders and Products**.**

·**Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.**

All entities' UIs include delete options. As mentioned above the Orders and OrderDetails table are joined and deleting an order from the Orders table will delete the row from the OrderDetails tables.

·**Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?**

Yes, update option is available for all entities.

·**Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.**

Yes, the Productions and Studios relationship is NULLable. Also the Orders/Productions, Orders/termsCodes, and Orders/SalesReps relationships are NULLable.

·**Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.**

Overall, the UI is very well designed and easy to navigate.

I suggest for the INSERT (add) to add a drop down menu that populates with the names of the FKs they reference instead of having the user entering a FKs value in the add form. For example, if the user wants to add a new production, the StudioID field should be Studio Name with a drop down menu that includes all studio names in the database. When the user selects a studio name, the name should refer to the studioID in the backend.

Also, you might want to check if joining two tables for the SELECT requirement is acceptable. You have both the Orders and the OrderDetails joined in one page under Orders.

Sara Biebuyck

- ·***Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is***

generally not acceptable for just a single query to join all tables and displays them.

- Yes, the UI utilizes SELECT for every table in their schema!

- ***·Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?***
  - Yes, there are more than one. Show All Productions By Studio, Show All Orders by Production, and Show all Products by Vendor, etc. are all available as options to filter their corresponding tables. That being said, I do this all these titles are a little wordy, and might benefit from condensing the names. Simplify to "Show by Production" or "Show by Studio" would be better, in my opinion. That being said, this is more that a foundation, most of the heavy-lifting front end has been done!
- ***·Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.***
  - Yes, the UI utilizes an INSERT for every table in schema.
- ***·Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).***
  - Yes! Each INSERT does add the FKs and one M:M relationship.
- ***·Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.***
  - Yes, in fact, DELETE is present on all tables. CASCADE is available as well, however,
- ***·Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?***
  - Yes, in fact, UPDATE is there for every table.
- ***·Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.***
  - Yes, multiple in fact. Productions/Studios, Orders/Productions, Orders/SalesReps, etc. Very in depth project!
- ***·Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.***

    - To be honest, not really. Most my suggestions are nit-picky. I think that the text fields could benefit from implementing margins, as is they are spread too far, and some fields are separated from their text box, I.E "City:" and it's corresponding textbox are on two different lines.
    - Additionally, I would add a column title for both "edit" and "delete", but again, that's just me being nitpicky.

Overall, amazing job!! I love your product and it has been implemented very thoroughly.

Gunar Turaids

- *Does the UI utilize a SELECT for every table in the schema?* In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

Yes, you do have SELECT queries for each page.

- *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*

You do have a search by functionalities (e.g. Prioducts by Vendor). There is no generalized search feature though (e.g. searching for a specific table item by entering an attribute name like a search for “Zaxel” in Vendors.). I’ll need to reread the project guide to be sure on the requirement.

- *Does the UI implement an INSERT for every table in the schema?* In other words, there should be UI input fields that correspond to each table and attribute in that table.

Yes, each entity appears to have an INSERT query and each page displays a form to Add. Look out in your SQL when defining for future input.

INSERT INTO `TermsCodes` (`TermCode`, `TermName`)

VALUES ('TermCode_input', 'TermName_input');

The VALUES should be identified by a special character.

- *Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?* In other words if there is a M:M relationship between Orders and

Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

Yes, insert queries include FKs defined in your DDL.

- *Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?* In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

Looks like your opting for DELETE on each entity. I also saw that you have ON DELETE SET NULL operations to prevent unwanted deletions. I had some trouble implementing this myself, but I think I understand the issue. Not far off from what you did.

- *Is there at least one UPDATE for any one entity?* In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

Yes, you do have some UPDATE queries. It appears that the HTML doesn't do anything yet. That is, clicking "Edit" does not bring up a new form.

I see that you opted for ON UPDATE NO ACTION, so your updates may not cascade to other tables. Perhaps this is intentional.

- *Is at least one relationship NULLable?* In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.

Yes. For example, in orders, you have the foreign key salesRepID which is NULLable.

- *Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.*

Nice favicon. Overall, the website looks very sharp and readable.

Does this need a special character?

WHERE orderID = orderID_from_creation_form

(i.e. :orderID_from_creation_form). There are some other examples of this. This is from the update orders query.

Mark Jordan:

Does the UI utilize a SELECT for every table in the schema? **In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.**

Yes, a SELECT query exists for each of the pages.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

Yes there were three search/filters that I counted.

Does the UI implement an INSERT for every table in the schema? **In other words, there should be UI input fields that correspond to each table and attribute in that table.**

Yes, the UI utilizes an INSERT for every table in the schema.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? **In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).**

Each INSERT adds the corresponding FK attributes, including at least one M:M relationship.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? **In other words, if an order is deleted from the Orders table, it**

should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

There is a DELETE statement for each table and one DELETE removes things from a M:M relationship.

***Is there at least one UPDATE for any one entity?* In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?**

There are several UPDATE queries but this not yet implemented in the UI.

***Is at least one relationship NULLable?* In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.**

Yes, the Production and Studios relationship is NULLable among others.

Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.

UI looks good! I feel like I need to improve mine now haha. Some suggestions:

- You could include drop down's in each of the input textfields for each of the add record pages to make each insertion consistent with previous insertions. For example, for the Studios form, since you've already provided a dropdown of Studios to show all Productions, using a dropdown field in the input form could help keep the naming of the record consistent e.g. all records for 'Netflix Productions' are listed as 'Netflix Productions' instead of 'Netflix', 'netflix', 'netflix pro', etc.

Actionable Improvement Steps (Step 3)

- Formatted table headers for legibility
- Form fields that reference a FK are dynamically populated by appropriate available FK's in the database via dropdown selector; i.e user does not manually enter FK values.

- Rewrote some SQL queries for proper functionality

FIFTYFIVE Sales and Service

Since 2020 there has been an explosion of new media platforms and more feature film work than ever. This year alone the film industry is expected to generate $95.45 billion in revenue. Economists conservatively estimate the industry will grow 7.2% annually over the next 8 years to an estimated $169.68 billion. In the last six months alone major studios such as Apple, Netflix, Paramount, and Disney have invested tens of billions of dollars in media acquisitions. Their 2022 production schedules are more than triple that of 2021. This is on the heels of the 2020 COVID-19 Pandemic that saw many companies that service the entertainment industry close their doors permanently. The extra demand from major entertainment companies coupled with the reduced supply caused by the COVID-19 pandemic has created a need for much more efficient tracking and business operation than ever before.

FIFTYFIVE Sales and Service is a one stop shop for film and television production companies with over $55 million in annual sales. Whether a student film or a major studio feature film, musical, episodic, or documentary we're here to help outfit your production with everything you could possibly need. Utilizing our proprietary database we are able to track orders, which production they are for, and from which studio the order came from. We also have the ability to track order details including the sales rep assigned to the order, the products in the order, and which vendor those products come from. These database abilities allow us to efficiently track tens of thousands of individual production orders, over 250,000 unique products, over 500 US-based vendors, and an entire sales staff, allowing us to fulfill each studio's needs quickly and accurately.

Database Outline

- **SalesReps**: records the Sales Representative who made the Orders for the Production.
    - salesRepID: PK, INT(11), Not Null, Auto-Increment
    - salesRepName: VARCHAR(45), Not Null
    - salesRepEmail: VARCHAR(45), Not Null
    - **Relationship**: 1:M relationship between **SalesReps** and **Orders** with *salesRepID* as a FK in **Orders**

- **Orders**: records orders placed by different Productions.
    - orderID: PK, INT(11), Not Null, Auto-Increment
    - productionID: FK, INT(11), Null
    - termsCodeID: FK, VARCHAR(45), Null
    - salesRepID: FK, INT(11), Null

    - orderDate: DATETIME, Not Null
    - purchaseOrder: VARCHAR(45), Not Null
    - CONSTRAINT `ProductionID` FOREIGN KEY (`productionID`) REFERENCES `Productions` (`productionID`) ON DELETE SET NULL ON UPDATE NO ACTION.
    - CONSTRAINT `SalesRepID` FOREIGN KEY (`salesRepID`) REFERENCES `SalesReps` (`salesRepID`) ON DELETE SET NULL ON UPDATE NO ACTION
    - CONSTRAINT `TermsCodeID` FOREIGN KEY (`termsCodeID`) REFERENCES `TermsCodes` (`termsCodeID`) ON DELETE NO ACTION ON UPDATE NO ACTION
    - **Relationship**: M:1 relationship between **Orders** and **Productions** with *productionID* as a FK in **Orders**. M:1 relationship between **Orders** and **SalesReps** with *salesRepID* as a FK in **Orders**. M:1 relationship between **Orders** and **TermsCode** with *termsCodeID* as a FK in **Orders**. M:M relationship between **Orders** and **Products**, implemented with the intersection table **OrderDetails**.

- **Products**: records all Products available to all Orders.
    - productID: PK, INT(11), Not Null, Auto-Increment
    - productName: VARCHAR(11), Not Null
    - cost: DECIMAL(19,2), Not Null
    - retailPrice: DECIMAL(19,2), Not Null
    - vendorID: FK, INT(11), Null
    - CONSTRAINT `VendorID` FOREIGN KEY (`vendorID`) REFERENCES `Vendors` (`vendorID`) ON DELETE SET NULL ON UPDATE NO ACTION
    - **Relationship**: M:M relationship between **Products** and **Orders**, implemented with the intersection table **OrderDetails**. M:1 relationship between **Products** and **Vendors** with *vendorID* as a FK in **Products**.

- **OrderDetails**: an intersection table that records all occurrences of Products and Orders, including order quantities and total amount.
    - orderDetailsID: PK, INT(11), Not Null Auto-Increment
    - orderID: FK, INT(11), Not Null
    - productID: FK, INT(11), Not Null
    - orderQty: INT(11), Not Null
    - unitPrice: DECIMAL(19,2), Not Null
    - totalAmount: DECIMAL(19,2), Not Null
    - CONSTRAINT `OrderID` FOREIGN KEY (`orderID`) REFERENCES `Orders` (`orderID`) ON DELETE CASCADE ON UPDATE NO ACTION,
    - CONSTRAINT `ProductID` FOREIGN KEY (`productID`) REFERENCES `Products` (`productID`) ON DELETE NO ACTION ON UPDATE NO ACTION
    - **Relationship**: M:1 relationship between **OrderDetails** and **Orders**, with *orderID* as a FK in **OrderDetails**. M:1 relationship between **OrderDetails** and **Products** with *productID* as a FK in **OrderDetails**.

- **Vendors**: records all Vendors for all Products available to all Orders.
    - vendorID: PK, INT(11), Not Null Auto-Increment
    - vendorName: VARCHAR(45), Not Null
    - contactName: VARCHAR(45), Not Null
    - contactEmail: VARCHAR(256), Not Null Unique
    - addressLine1: VARCHAR(50), Not Null
    - addressLine2: VARCHAR(50), Null
    - city: VARCHAR(45), Not Null
    - state: VARCHAR(45), Not Null
    - zipCode: VARCHAR(45), Not Null
    - **Relationship**: 1:M relationship between **Vendors** and **Products** with *vendorID* as a FK in **Products**.

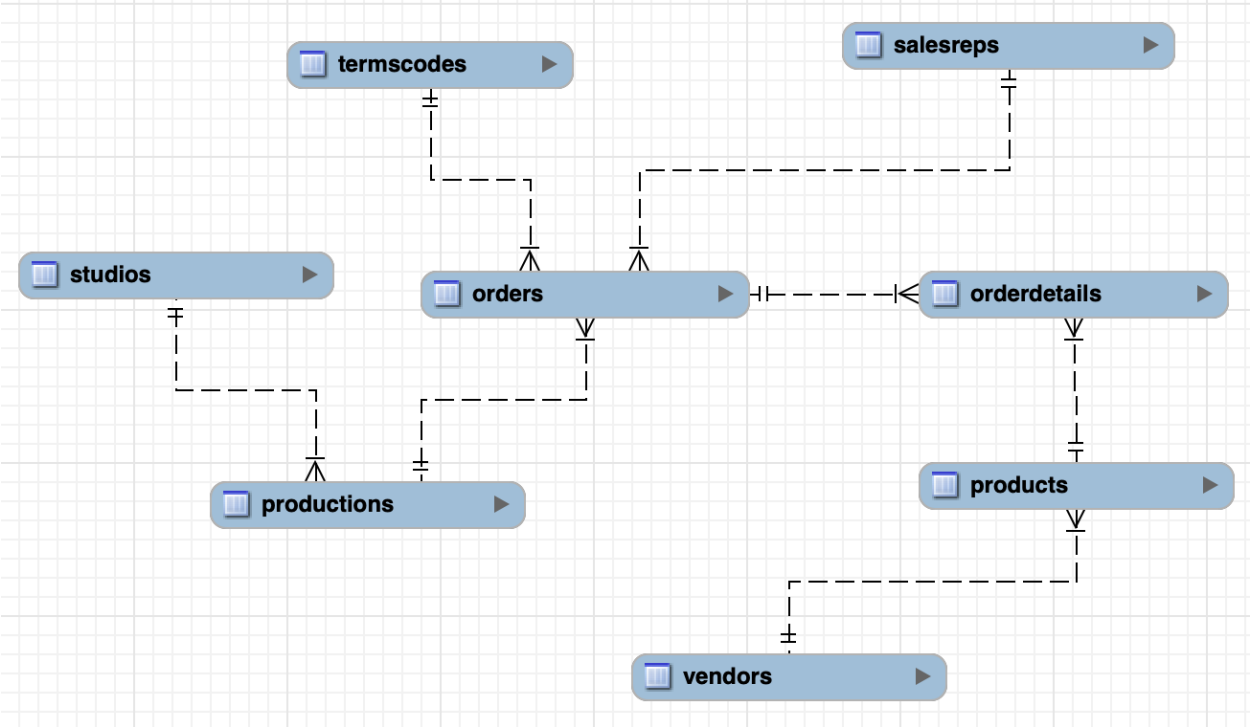
- **Productions**: records details of all film and television Productions that do business with us
    - productionID: PK, INT(11), Not Null, Auto-Increment
    - studioID: FK, INT(11), Null
    - showName: VARCHAR(45), Not Null
    - contactName: VARCHAR(45), Not Null
    - contactEmail: VARCHAR(256 Not Null Unique
    - addressLine1: VARCHAR(50), Not Null
    - addressLine2: VARCHAR(50), Not Null
    - city: VARCHAR(45), Not Null
    - state: VARCHAR(45), Not Null
    - zipCode: VARCHAR(45) Not Null
    - CONSTRAINT `StudioID` FOREIGN KEY (`studioID`) REFERENCES `Studios` (`studioID`) ON DELETE SET NULL ON UPDATE NO ACTION
    - **Relationship**: 1:M relationship between **Productions** and **Orders**, with *productionID* as a FK in **Orders**. 1:M relationship between **Studios** and **Productions**, with *studioID* as a FK in **Productions**.

- **Studios**: records details of all Studios that produce different Productions that in turn do business with us.
    - studioID: PK, INT(11), Not Null, Auto-Increment
    - studioName: VARCHAR(45), Not Null
    - contactName: VARCHAR(45), Not Null
    - contactEmail: VARCHAR(256), Not Null Unique
    - addressLine1: VARCHAR(50), Not Null
    - addressLine2: VARCHAR(50), Null
    - city: VARCHAR(45), Not Null
    - state: VARCHAR(45), Not Null
    - zipCode: VARCHAR(45) Not Null

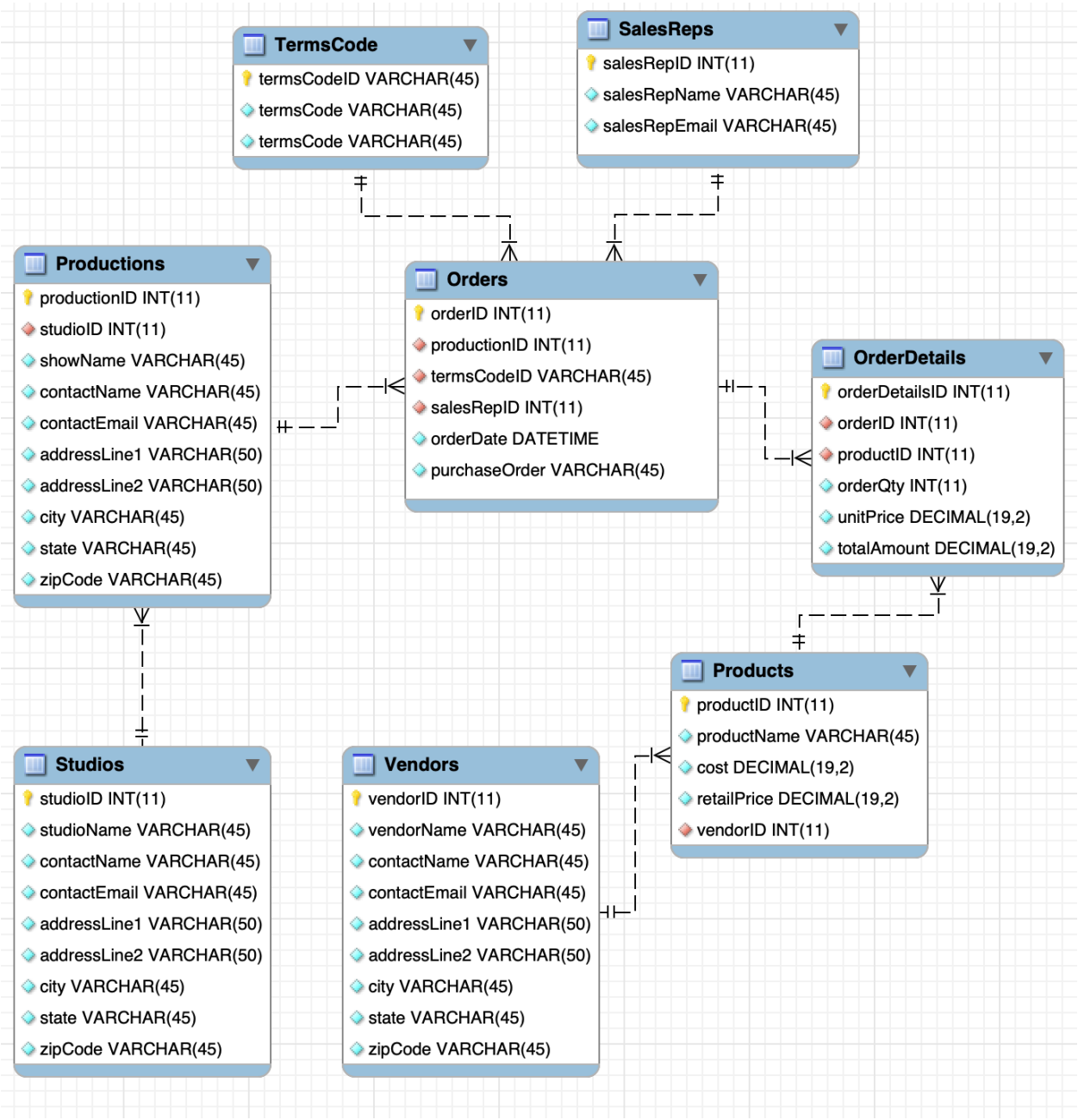
    - **Relationship**: 1:M relationship between **Studios** and **Productions** with *studioID* as a FK in **Productions**.

- **TermsCodes**: records all payment Terms available to Orders.
    - termsCodeID: PK, VARCHAR(45), Not Null Auto-Increment
    - termCode: VARCHAR(45) Not Null
    - termName: VARCHAR (45), Not Null
    - **Relationship**: 1:M relationship between **TermsCode** and **Orders** with *termsCodeID* as a FK in **Orders**.

Entity Relationship Diagram

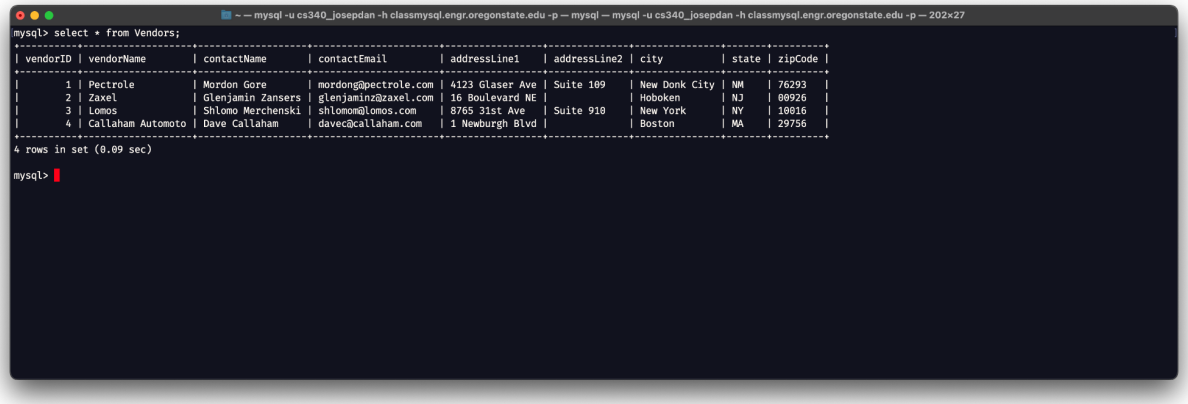


Schema

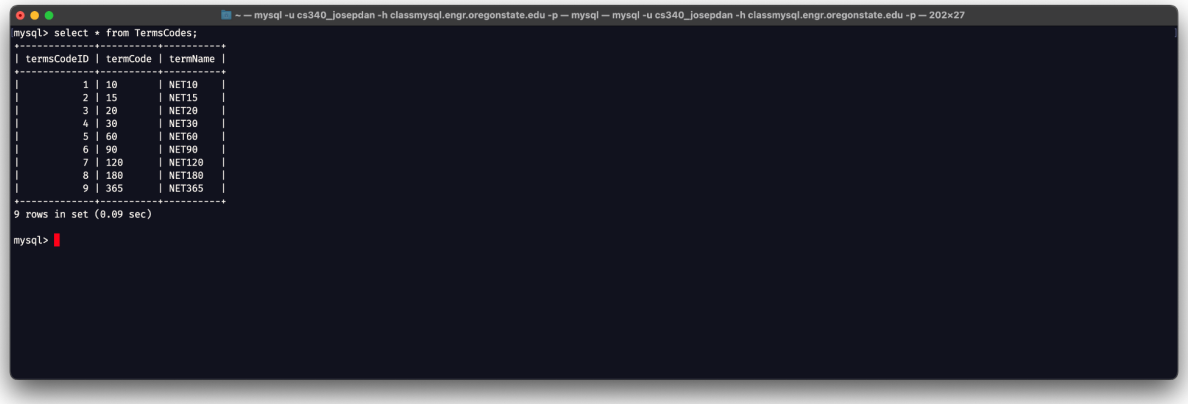

TermsCode
termsCodeID VARCHAR(45)
termsCode VARCHAR(45)
termsCode VARCHAR(45)
SalesReps
salesRepID INT(11)
salesRepName VARCHAR(45)
salesRepEmail VARCHAR(45)
Productions
productionID INT(11)
studioID INT(11)
showName VARCHAR(45)
contactName VARCHAR(45)
contactEmail VARCHAR(45)
addressLine1 VARCHAR(50)
addressLine2 VARCHAR(50)
city VARCHAR(45)
state VARCHAR(45)
zipCode VARCHAR(45)
Orders
orderID INT(11)
productionID INT(11)
termsCodeID VARCHAR(45)
salesRepID INT(11)
orderDate DATETIME
purchaseOrder VARCHAR(45)
OrderDetails
orderDetailsID INT(11)
orderID INT(11)
productID INT(11)
orderQty INT(11)
unitPrice DECIMAL(19,2)
totalAmount DECIMAL(19,2)
Products
productID INT(11)
productName VARCHAR(45)
cost DECIMAL(19,2)
retailPrice DECIMAL(19,2)
vendorID INT(11)
Studios
studioID INT(11)
studioName VARCHAR(45)
contactName VARCHAR(45)
contactEmail VARCHAR(45)
addressLine1 VARCHAR(50)
addressLine2 VARCHAR(50)
city VARCHAR(45)
state VARCHAR(45)
zipCode VARCHAR(45)
Vendors
vendorID INT(11)
vendorName VARCHAR(45)
contactName VARCHAR(45)
contactEmail VARCHAR(45)
addressLine1 VARCHAR(50)
addressLine2 VARCHAR(50)
city VARCHAR(45)
state VARCHAR(45)
zipCode VARCHAR(45)

Example Data

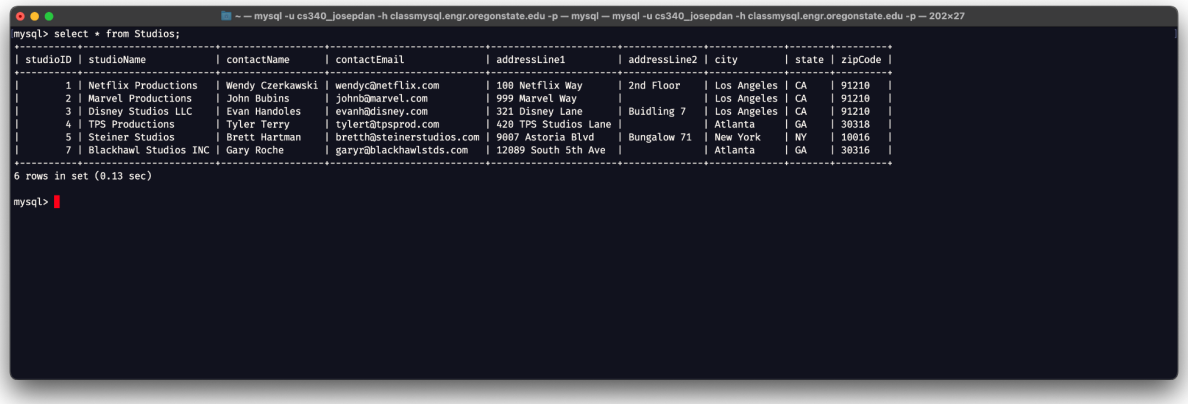
Vendors:



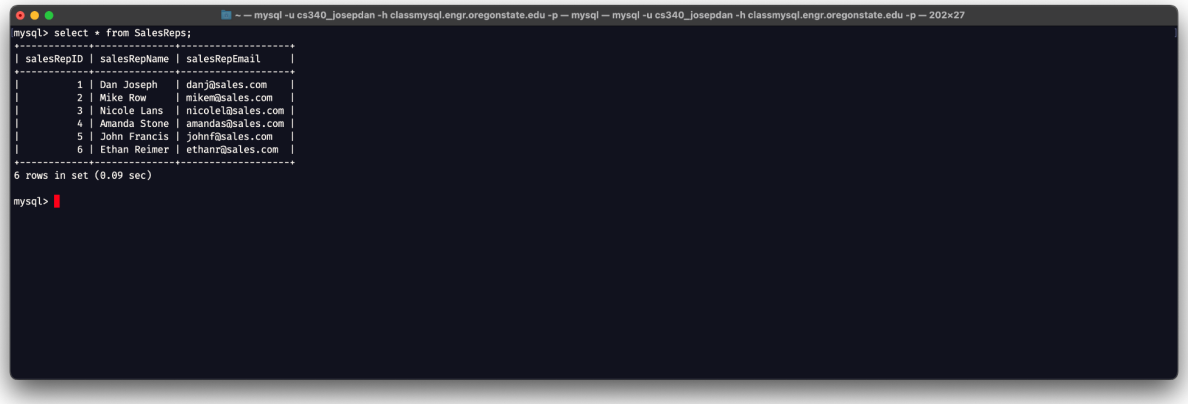
TermsCode:



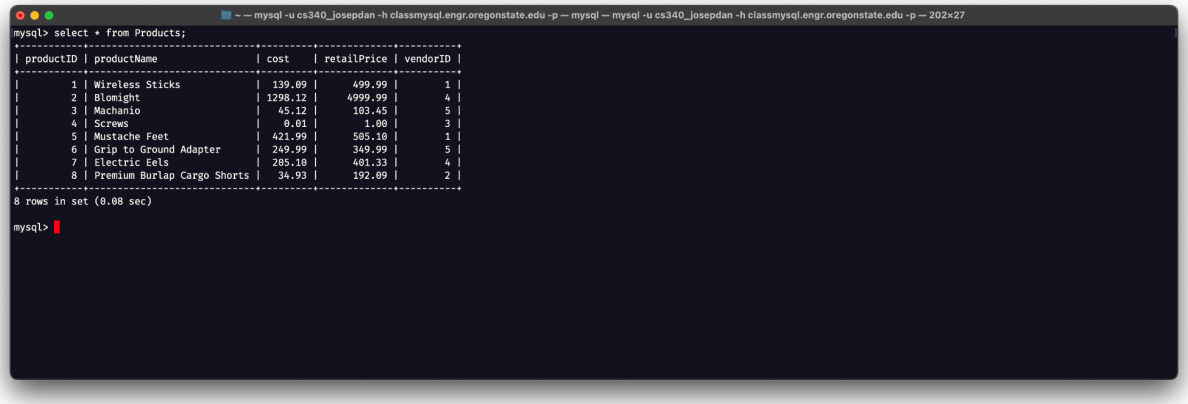
Studios:



SalesReps:



Products:



Productions:

~ — mysql -u cs340_josepdan -h classmysql.engr.oregonstate.edu -p — mysql — mysql -u cs340_josepdan -h classmysql.engr.oregonstate.edu -p — 202×27

```
mysql> select * from Productions;
+--------------+----------+-----------------------------------------+--------------+--------------------+-----------------------+-----------------------+---------------+-------+---------+
| productionID | studioID | showName                                | contactName  | contactEmail       | addressLine1          | addressLine2          | city          | state | zipCode |
+--------------+----------+-----------------------------------------+--------------+--------------------+-----------------------+-----------------------+---------------+-------+---------+
|            1 |        3 | Great Baking Show                       | Abdul Rena   | abdulr@netflix.com | 89 Peachtree Tr.      |                       | Atlanta       | GA    | 30316   |
|            2 |        3 | Real Housedogs Season 24                | Butch Uppercut | butchu@netflix.com | 345 Blackhall Studio Way | Building 43        | Atlanta       | GA    | 30031   |
|            3 |        4 | Avengers 17: How Hawkeye Got His Groove Back | Lance Steel | lances@marvel.com  | 1001 Veterans Pkwy    | Building 2, 3rd Floor | Los Angeles   | CA    | 90210   |
|            4 |     NULL | Madeas Nose Job                         | Pyler Terry  | pylert@tpsprod.com | 123 TPS Way           |                       | Altanta       | GA    | 30103   |
|            5 |     NULL | Madea vs Gout                           | Myler Sterry | mylers@tpsprod.com | 123 TPS Way           |                       | Atlanta       | GA    | 30103   |
|            6 |        5 | Sneak King X Glee                       | Madeline Kury | madelinek@disney.com | 2013 East 89th Ave | Suite 721             | New York      | NY    | 10019   |
|            7 |        1 | Holy Moly Tony!                         | Chad Winkle  | chadw@netflixprod.com | 124 Gastron Blvd   | Trailer 4             | Salt Lake City | UT   | 12903   |
+--------------+----------+-----------------------------------------+--------------+--------------------+-----------------------+-----------------------+---------------+-------+---------+
7 rows in set (0.09 sec)

mysql>
```

Orders:

~ — mysql -u cs340_josepdan -h classmysql.engr.oregonstate.edu -p — mysql — mysql -u cs340_josepdan -h classmysql.engr.oregonstate.edu -p — 202×27

```
mysql> select * from Orders;
+---------+--------------+--------------+------------+---------------------+---------------+
| orderID | productionID | termsCodeID | salesRepID | orderDate           | purchaseOrder |
+---------+--------------+--------------+------------+---------------------+---------------+
|       1 |            1 |            1 |          2 | 2022-07-08 22:08:15 | 1903          |
|       2 |            1 |            1 |          2 | 2022-06-12 22:08:45 | 12984         |
|       3 |            4 |            2 |          4 | 2022-07-08 22:11:32 | 14A12         |
|       4 |            5 |            3 |          1 | 2022-07-09 17:17:14 | A103          |
|       5 |            5 |            4 |          1 | 2022-07-09 17:17:14 | A104          |
|       6 |            6 |            5 |          4 | 2022-07-09 17:18:58 | 2148          |
|       7 |            6 |            6 |          5 | 2022-07-09 17:18:58 | 1210          |
|       8 |            7 |            4 |          6 | 2022-07-18 00:36:06 | A098          |
+---------+--------------+--------------+------------+---------------------+---------------+
8 rows in set (0.09 sec)

mysql>
```

OrderDetails:

~ — mysql -u cs340_josepdan -h classmysql.engr.oregonstate.edu -p — mysql — mysql -u cs340_josepdan -h classmysql.engr.oregonstate.edu -p — 202×27

```
mysql> select * from OrderDetails;
+----------------+---------+-----------+----------+-----------+-------------+
| orderDetailsID | orderID | productID | orderQty | unitPrice | totalAmount |
+----------------+---------+-----------+----------+-----------+-------------+
|              1 |       1 |         2 |        1 |     99.00 |       99.00 |
|              2 |       3 |         1 |        2 |     40.00 |       80.00 |
|              3 |       2 |         4 |      100 |      1.00 |      100.00 |
|              4 |       4 |         6 |        1 |    349.00 |      349.00 |
|              5 |       4 |         3 |        2 |    200.00 |      400.00 |
|              6 |       5 |         5 |        1 |    124.00 |      124.00 |
|              7 |       6 |         6 |        1 |    400.00 |      400.00 |
|              8 |       7 |         5 |        2 |    200.00 |      400.00 |
|              9 |       7 |         3 |        2 |     11.00 |       22.00 |
|             10 |       8 |         7 |       12 |    302.99 |     3635.88 |
|             11 |       8 |         4 |      100 |      1.00 |      100.00 |
+----------------+---------+-----------+----------+-----------+-------------+
11 rows in set (0.10 sec)

mysql>
```